

# Upotreba alata Apache Jmeter

U okviru ove lekcije govorićemo o ranije spomenutom alatu Apache JMeter. Objasnićemo njegovu svrhu, kako se koristi i kako postizemo automatizaciju stress testiranja i kreiranje skripti za testove.

Apache JMeter je alat za automatsko load i stress testiranje. Pored toga, ovim alatom se može pratiti funkcionalno ponašanje programa i vršiti precizno merenje performansi programa. Alat je kreiran pomoću programskog jezika Java i open source je softver.

Originalna svrha alata je bilo testiranje web aplikacija ili FTP aplikacija. Danas se koristi za stress, load, functional i database testiranje, kao i mnoge druge vidove testiranja. Za razliku od jednostavnih alata poput Apache Bench, može se testirati kroz različite protokole, uključujući i HTTPS i SMTP protokole.



Slika 13.1. Apache JMeter

Za razliku od drugih rešenja ovog tipa, može se koristiti u konzolnom i GUI režimu. Dakle, imamo odvojeno radno okruženje u okviru kojeg pokrećemo i pišemo testove.

## Prednosti korišćenja JMeter alata:

- open source licenca – dakle, alat je potpuno besplatan;
- jednostavan interfejs – za korisnike koji imaju iskustvo rada u razvojnim okruženjima, upoznavanje sa programom je brzo i lako;
- vizuelizacija testiranja alata omogućava prikaz rezultata u vidu tabela, dijagrama, log fajlova i grafikona;
- simulacija korisnika – JMeter omogućava simuliranje korisnika koji šalju zahteve ka aplikaciji na odvojenim nitima, što dodatno omogućava kreiranje realnih uslova prilikom opterećenja aplikacije;
- platformski nezavisan – kako je program napisan u Java programskom jeziku, može raditi na više platformi; postoji podrška za operativne sisteme Windows, macOS i Linux;
- testiranje skripti – JMeter može biti povezan i sa alatom Selenium u svrhu postizanja automatskog testiranja.

Nakon ovog kratkog uvoda, počinjemo sa radom. Prvi korak je, naravno, instalacija samog programa. On se može preuzeti sa sledećeg linka:

[https://jmeter.apache.org/download\\_jmeter.cgi](https://jmeter.apache.org/download_jmeter.cgi)

Prilikom preuzimanja, prema zvaničnoj preporuci, uvek preuzmite najnoviju verziju programa.

### Napomena

Kako je JMeter Java aplikacija, za njeno pokretanje i korišćenje je potrebno imati instaliran Java programski paket, koji omogućava podršku za Java aplikacije na vašem računaru.

Stoga, u slučaju da se prilikom instalacije alata prikaže greška u vezi sa Java verzijom, potrebno je preuzeti je sa sledećeg linka:

**<https://www.java.com/en/download/manual.jsp>**

U okviru Binaries sekcije, preuzimamo zip fajl ovog alata (slika 13.2):

## Download Apache JMeter



We recommend you use a mirror to download our release builds, but you **must** [verify the integrity](#) of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from all the mirrors.

You are currently using <https://downloads.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

The **KEYS** link links to the code signing keys used to sign the product. The **PGP** link downloads the OpenPGP compatible signature from our main site. The **SHA-512** link downloads the sha512 checksum from the main site. Please [verify the integrity](#) of the downloaded file.

For more information concerning Apache JMeter, see the [Apache JMeter](#) site.

[KEYS](#)

## Apache JMeter 5.x.x (Requires Java 8+)

### Binaries

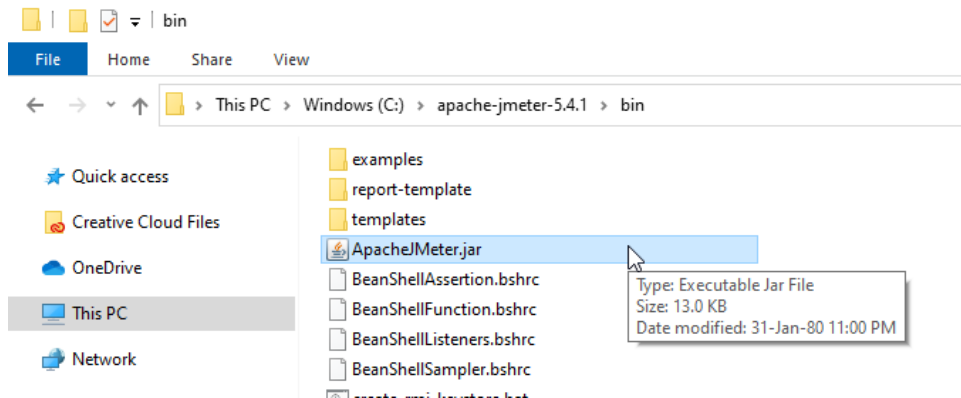
[apache-jmeter-5.x.x.tar.gz](#) [sha512](#) [pgp](#)  
[apache-jmeter-5.x.x.zip](#) [sha512](#) [pgp](#)

### Source

*Slika 13.2. Preuzimanje zip fajla alata JMeter*

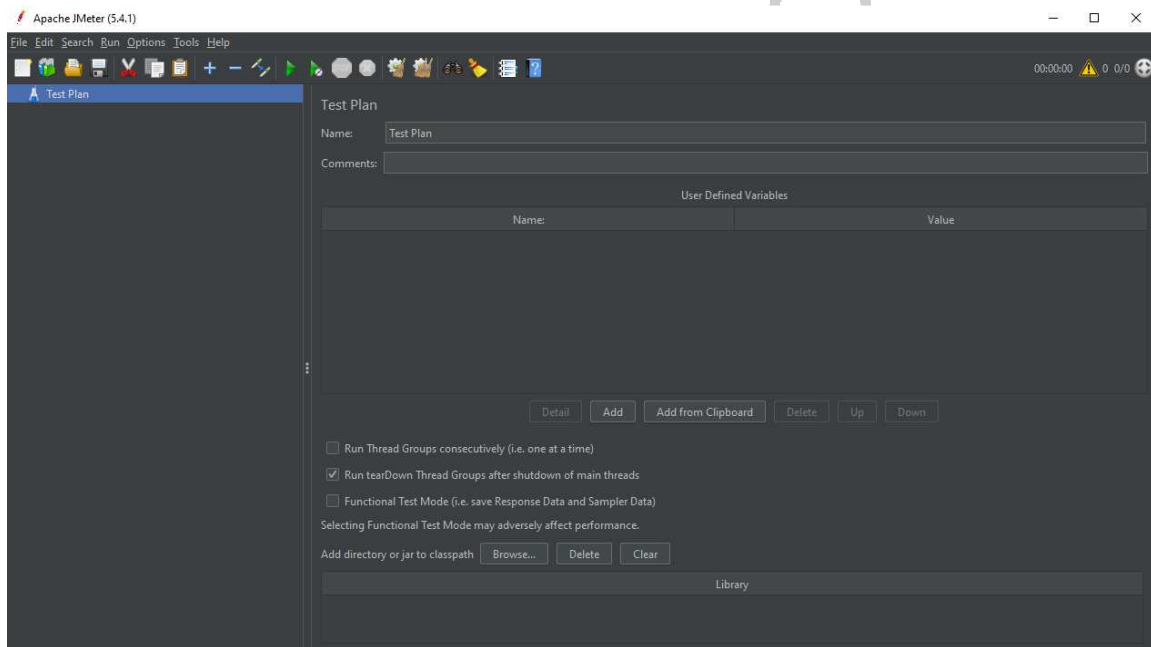
Nakon preuzimanja zip fajla, potrebno je raspakovati arhivu na željenu lokaciju. U ovom slučaju, lokacija na kojoj ćete raspakovati arhivu je potpuno proizvoljna, ali važno je da imate lak pristup folderu, s obzirom na to da se program pokreće direktno iz samog foldera.

Kada je arhiva raspakovana, program možete pokrenuti korišćenjem .jar fajla, koji nosi naziv kao i sam program – ApacheJMeter.jar. Ovaj fajl možemo pronaći u okviru **bin** foldera (slika 13.3).



Slika 13.3. Pokretački fajl unutar bin foldera

Otvaranjem ovog fajla, pokrećemo i sam program i dobijamo prikaz kao na slici u nastavku.

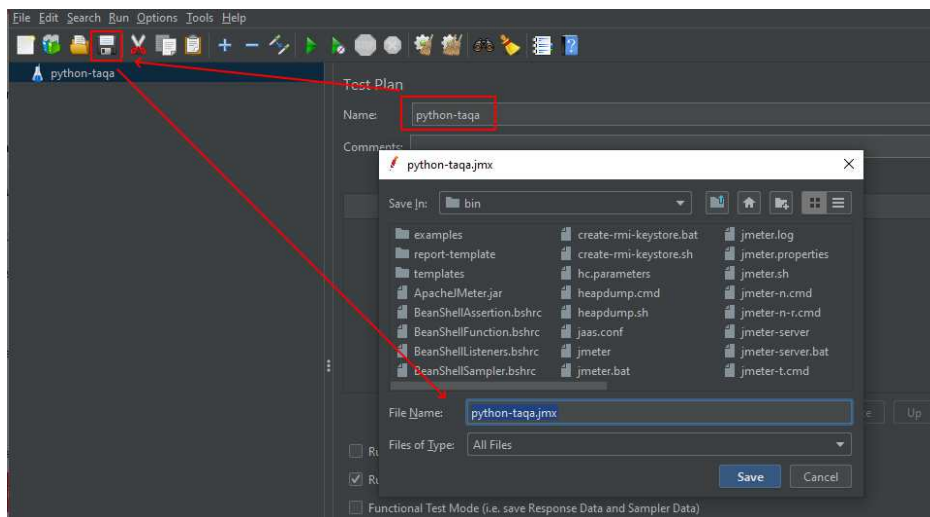


Slika 13.4. Radno okruženje alata JMeter

Upravo možete videti radno okruženje ovog alata. Dakle, imamo mali broj opcija, od kojih je samo par specifično za alate ovoga tipa, pa ćemo u nastavku lekcije govoriti o njima.

Prvi korak u radu sa ovim alatom jeste kreiranje test plana. Da bi JMeter test mogao da bude startovan, on mora postojati na fajl sistemu u formi test plana i, kao što možete videti, sa leve strane interfejsa programa se prikazuje struktura fajlova alata. Trenutno postoji samo jedna stavka i to je *Test Plan*, koji se podrazumevano kreira prilikom startovanja programa. Na nama je da sada promenimo naziv sa tog podrazumevanog i da ga sačuvamo.

Stoga u polju *Name* unosimo proizvoljan naziv za svoj test plan. Nakon toga ostaje samo da sačuvamo svoj test plan. Ovo možemo postići kao i kod drugih programa, izborom menija *File* i opcije *Save*, prečicom *Ctrl+S* ili opcijom prikazanom ikonicom diskete u samom interfejsu programa. Kako odaberemo da sačuvamo test plan, prikazuje se prozor prikazan na slici u nastavku.

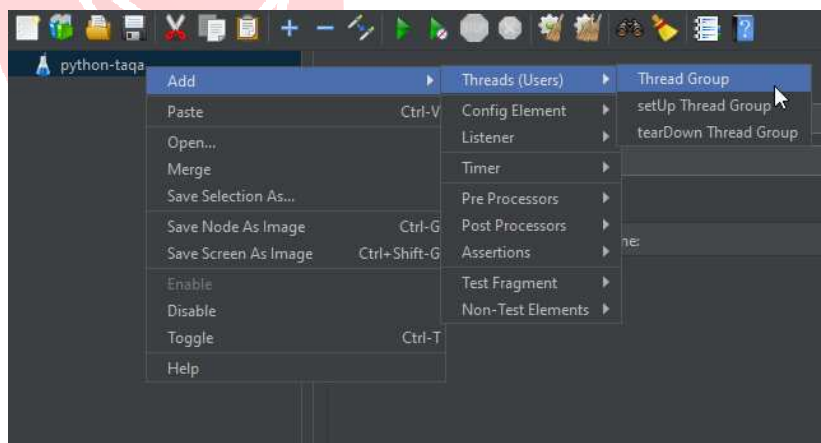


Slika 13.5. Čuvanje test plana u okviru alata JMeter

Test plan se čuva u .jmx formatu. Ovo je format specifičan za JMeter i možemo ga posmatrati kao radni fajl ili projekat u okviru JMeter alata. U okviru tog fajla će se čuvati svi napisani kodovi i parametri testa.

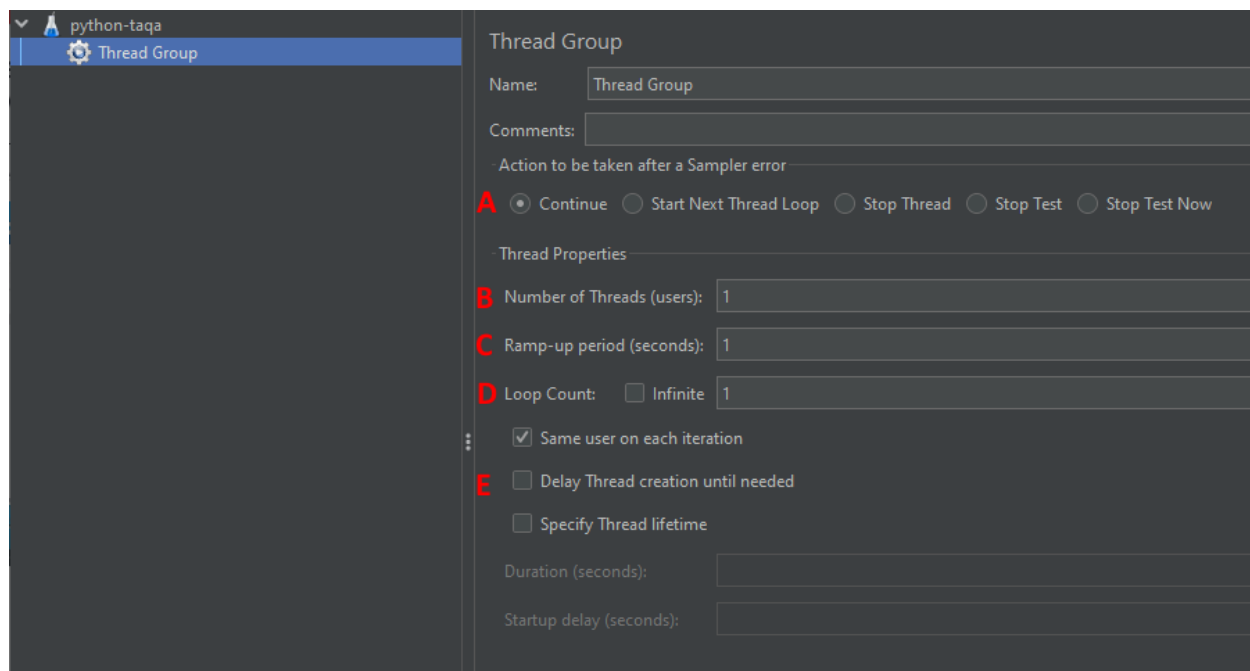
Kada imamo sačuvan test plan, naredni korak je da definišemo broj korisnika koji ćemo koristiti za testiranje, da li su korisnici uporedni, trajanje testova i slično. Sve korake i parametre jednog test plana dodajemo desnim klikom na sam test plan i izborom stavke *Add*.

U okviru same stavke imamo veći broj opcija, ali prethodno spomenute korisnike dodajemo biranjem opcije *Threads (Users)* i izborom stavke *Thread Group*.



Slika 13.6. Kreiranje grupe korisnika

Nakon izbora opcije dobijamo parametre za kreiranje Thread grupe. Unutar Thread grupe, nalaze se podešavanja o broju konkurentnih korisnika, periodu uhodavanja testa, broju iteracija testa i slično.



Slika 13.7. Parametri Thread grupe

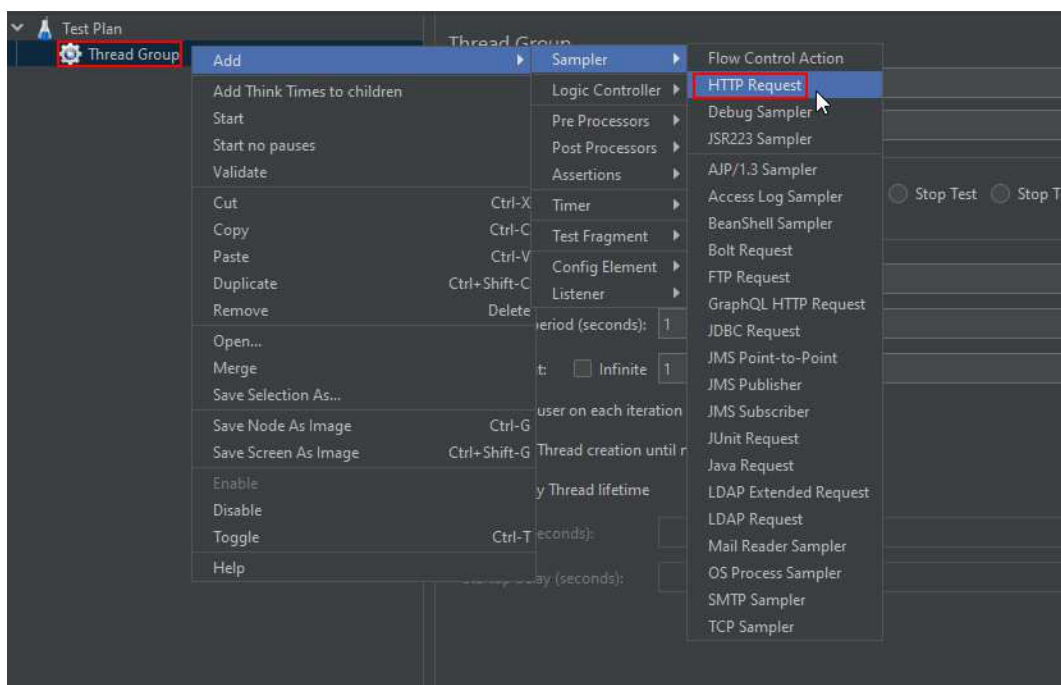
Pored osnovnih parametara, poput naziva grupe i opisa, imamo:

- A – izbor akcije koja će se izvršiti ukoliko dođe do greške prilikom izvršavanja niti;
- B – broj konkurentnih niti (korisnika);
- C – pauza između kreiranja niti;
- D – broj ponavljanja;
- E – podešavanja dodatnih parametara trajanja.

Kao što možete videti na slici 13.7, podrazumevana podešavanja Thread grupe su: jedan korisnik, period čekanja od jedne sekunde i simulirani korisnik radi nešto samo jednom.

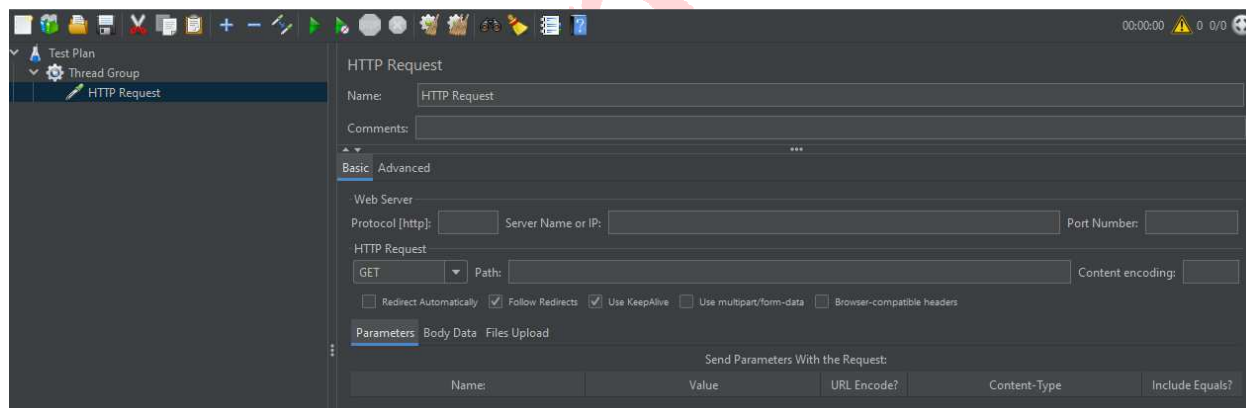
Kada kreiramo Thread grupu, tj. odredimo grupu simuliranih korisnika, sledeći korak je da definišemo šta konkretno naši korisnici rade. Ovo postizemo pomoću semplera.

Sempleri su različite, konkretne operacije koje JMeter izvršava u okviru jedne grupe niti. Sempleri su najvećim delom zahtevi ili neke druge vrste mrežnih operacija. Najkorišćeniji sempler je HTTP Request, pa ćemo njega i dodati u projekat. Semplere dodajemo nad Thread grupom; prethodno smo definisali parametre test plana, a sada definišemo parametre za grupu korisnika (niti).



Slika 13.8. Sampler podmeni i opcija HTTP Request

Kada izaberemo HTTP Request sampler, možemo videti deo interfejsa kao na slici prikazanoj u nastavku.



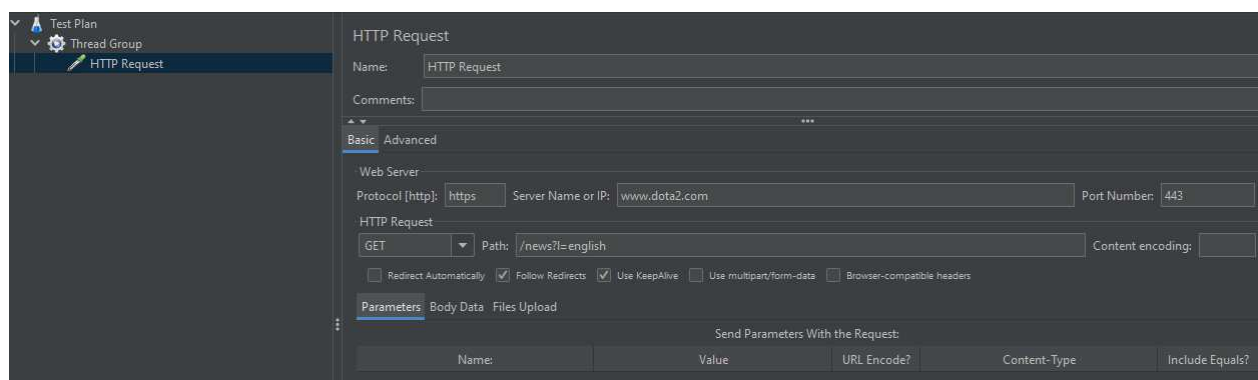
Slika 13.9. Parametri opcije HTTP Request

HTTP Request predstavlja jedan HTTP zahtev, pa stoga sada možemo definisati da će korisnik pristupiti nekom sajtu. Samo radi primera, koristićemo sajt popularne igre Dota 2. Link je sledeći: <https://www.dota2.com/news?l=english>

Sada je samo potrebno da parametre sajta unesemo unutar HTTP requesta.

Prvi važan parametar je protokol koji ćemo koristiti. Kako možemo videti, moramo koristiti HTTPS protokol, pa stoga i to unosimo u polje *Protocol*. Sledeće polje je *Server Name*, gde kucamo adresu samog sajta; u našem slučaju, to je [www.dota2.com](http://www.dota2.com). Kao broj porta (polje *Port Number*) unosimo 443, podrazumevani port HTTPS protokola. Kao tip zahteva ostavljamo GET, jer nam je potrebno samo dobavljanje sajta. Sledeći korak je *Path*. U okviru polja *Path* unosimo tačnu putanju do željene strane. Kako u našem linku možete videti, pristupamo vestima na sajtu sa izabranim jezikom, pa stoga u *Path* polje unosimo: /news?l=English.

Prikaz unetih parametara možete videti i na sledećoj slici:



Slika 13.10. Uneti parametri opcije HTTP Request

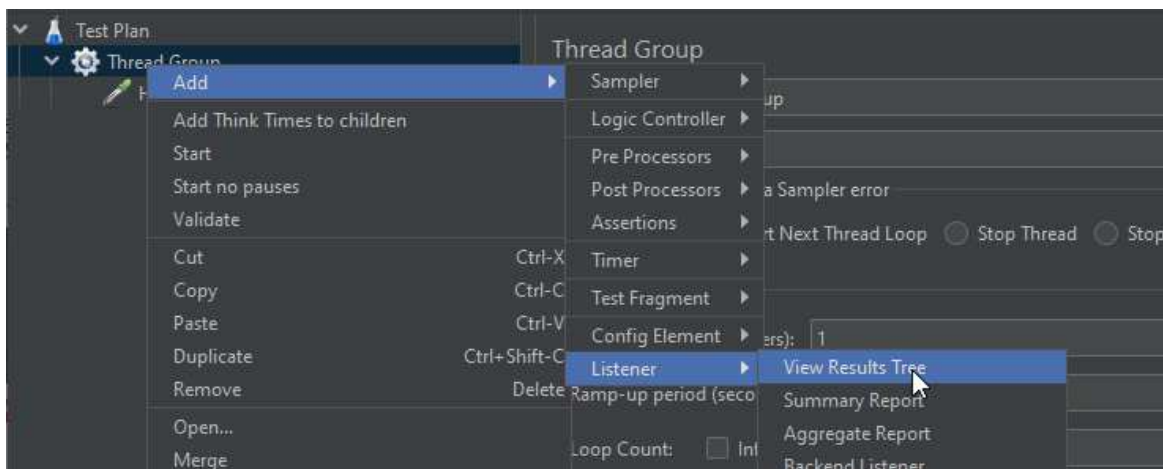
Ukoliko sačuvamo svoj napredak, sada smo definisali jedan jednostavan test, gde jedan korisnik pristupa sajtu. Ukoliko odaberemo pokretanje testa, izborom zelenog *play* tastera, videćemo da se ništa posebno neće dogoditi. Razlog toga je što je, za razliku od Apache Bencha, u okviru JMetra potrebno dodati posebnu komponentu koja se naziva slušač. Slušači nam, između ostalog, omogućavaju da pregledamo rezultate rada testa.

## JMeter slušači

Pomoću slušača, materijalizuju se rezultati izvršavanja semplera u okviru grupa niti. Slušači mogu vizualizovati rezultate, transformisati ih, ili ih prosledivati drugim sistemima (fajl sistemu, bazi podataka i slično).

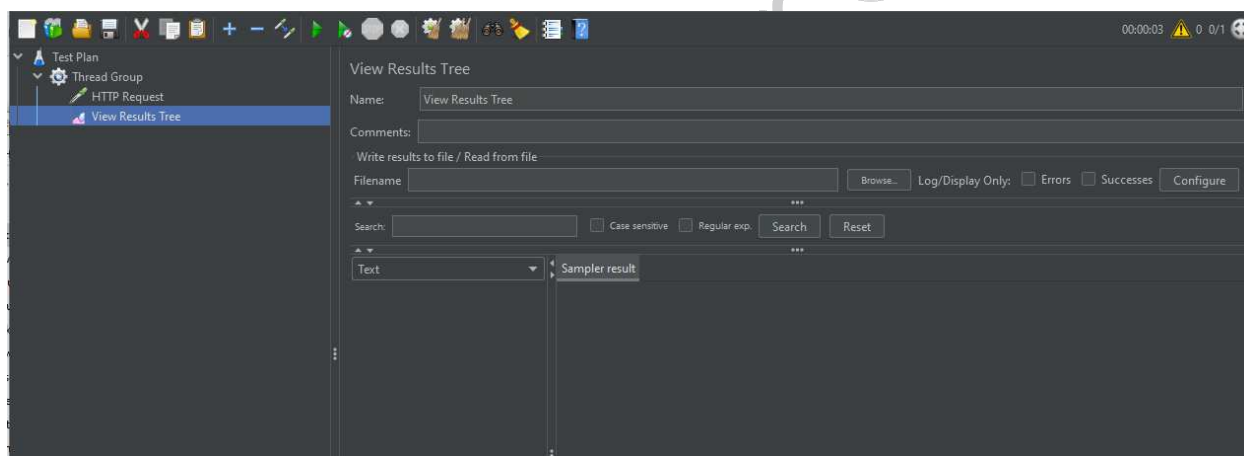
Slušač možemo dodati nad samim zahtevom (HTTP request) kada želimo da vidimo rezultate izvršavanja samog tog zahteva. Alternativno, možemo postaviti slušač nad celom Thread grupom i na taj način dobijamo pregled rezultata svih semplera.

Stoga koristimo već poznati *Add* meni i stavku *Listener*. U okviru JMetra postoji veliki broj slušača, u zavisnosti od toga koji vid aplikacije koristimo i na koji način želimo da prikažemo rezultate – kroz neki vid vizuelnog izveštaja, tabela ili jednostavno u okviru programa. U našem slučaju koristićemo listener koji se zove View Result Tree. Ovaj listener nam daje jednostavan i brz pregled u okviru JMetra.



Slika 13.11. Dodavanje slušača nad Thread grupom

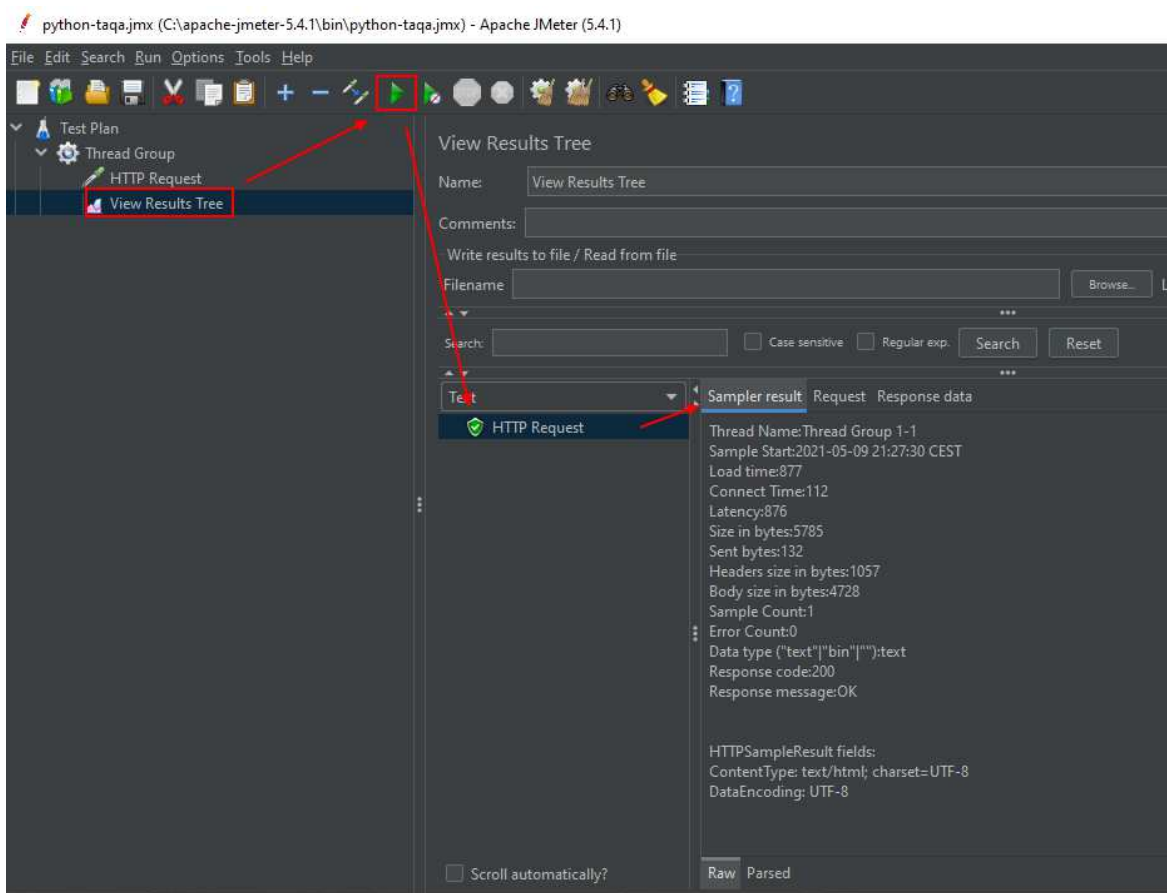
Nakon dodavanja slušača, možemo videti sledeći prikaz:



Slika 13.12. View Result Tree slušač

Sada smo definisali korisnika, definisali šta želimo da korisnik uradi i odabrali način da prikazemo rezultate. Ukoliko sada pokrenemo projekat, u okviru View Result Tree slušača videćemo da se izvršio HTTP request. Ukoliko kliknemo na ikonicu u *Sampler result* sekciji, vidimo i rezultate.





Slika 13.13. Prikaz rezultata izvršavanja zahteva

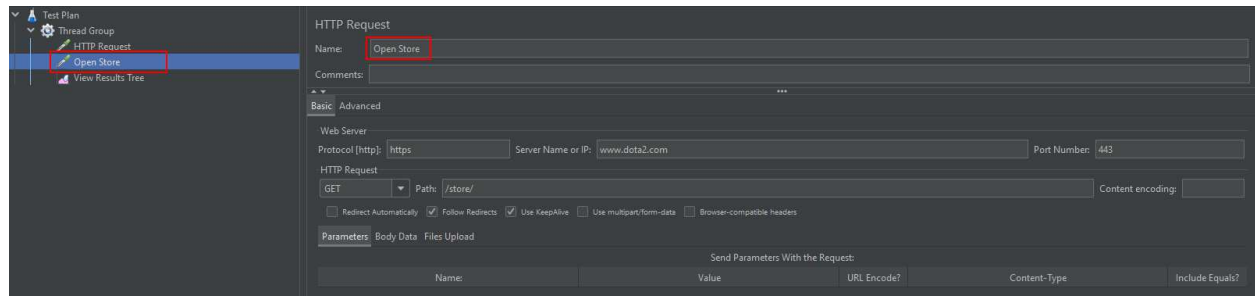
Sada u okviru rezultata samplera možemo videti neke parametre koji su nam od ranije poznati, poput vremena konektovanja, vremena učitavanja i odziva sajta, ali i neke detaljnije podatke poput veličine preuzetih paketa, koliko je podataka poslao JMeter, koji su statusni kodovi servera i slično. Ovi podaci se tiču samog testa, tj. HTTPS pristupa sajtu. Naravno, ovo nije load ili stress test, s obzirom na to da smo koristili samo jednog korisnika. Važan aspekt je kartica *Response data*. U okviru ove kartice dobijamo rezultat odgovora. U ovom slučaju, kako je ovo bio zahtev ka sajtu, kao odgovor smo dobili kompletan kod stranice. Dakle, JMeter simulira korisnika, ali ne simulira zahtev – za sajt je ovo bio jedan realan korisnik.

Kada smo govorili o Apache Benchu, rekli smo da je jedna od njegovih mana to što može da testira samo jednu stranicu u okviru jednog testa. Ovo nije slučaj kod Jmetera – sada možemo reći da korisnik nakon pristupa stranici News želi da pristupi sekciji Store. Na ovaj način sumiramo aktivnost korisnika na sajtu ili web aplikaciji. U okviru Jmetera ovo postizemo jednostavno još jednim HTTP requestom, samo ka drugoj stranici sajta; koristićemo link: <https://www.dota2.com/store/>.

Dakle, ponovo kroz Samplers sekciju dodajemo HTTP request i unosimo podatke za ovu novu stranicu:

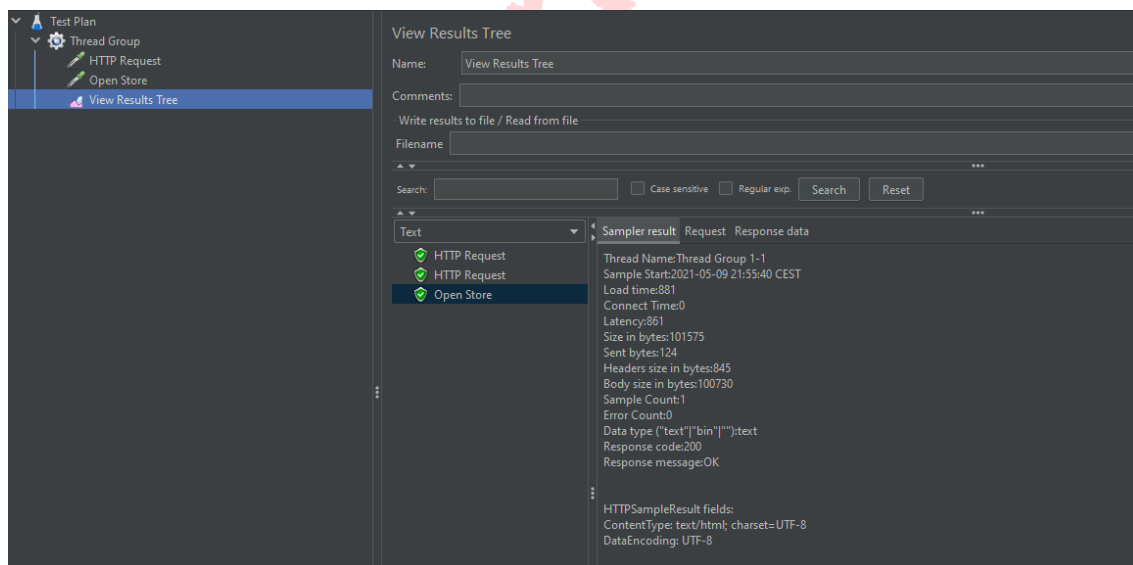
Protokol: https  
Server Name: www.dota2.com  
Port Number: 443  
Metod: GET  
Path: /store/

Dakle, možete videti da se samo putanja razlikuje. Za sam HTTP request promenićemo naziv da se razlikuje od prethodno dodatog; neka naziv bude *Open Store*. Da bismo dobili prikaz rezultata i za ovaj HTTP zahtev, potrebno je na levoj strani interfejsa prevući ga iznad *View Results Tree*.



Slika 13.14. Dodavanje zahteva

Ukoliko sada sačuvamo i pokrenemo naš projekat, u sekciji *View Results Tree* možemo videti izvršavanje više zahteva, od kojih će svaki imati detaljne informacije o vremenima odziva, veličinama paketa i kodu stranice.



Slika 13.15. Rezultat izvršavanja zahteva

Nakon ovog koraka, postigli smo da imamo definisane korisnike koji pristupaju nekoj stranici sajta i zatim odlaze na drugu stranicu i imamo način da izmerimo promene sa povećanjem broja korisnika. Broj korisnika jednostavno menjamo tako što uvećamo broj korisnika u samoj Thread grupi. Naravno, kao i kod Apache Bench, povećavamo broj korisnika sve dok ne dođe do grešaka ili prevelikog vremena odziva sajta.

Pored stress i load testova, u okviru JMetera možemo pisati Assertion – odnosno, proveriti da li je neki uslov ispunjen.

### Pitanje

Prikaz rezultata u okviru alata JMeter postižemo opcijom:

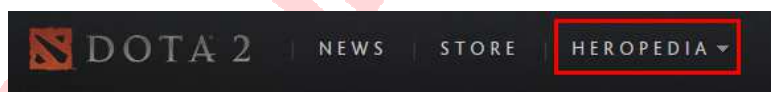
- **Listener**
- Sampler
- Thread

### Objašnjenje:

*Listener ili slušač služi za prikaz rezultata izvršavanja semplera u okviru grupa niti. Slušači mogu vizuelizovati rezultate, transformisati ih, ili ih prosledivati drugim sistemima.*

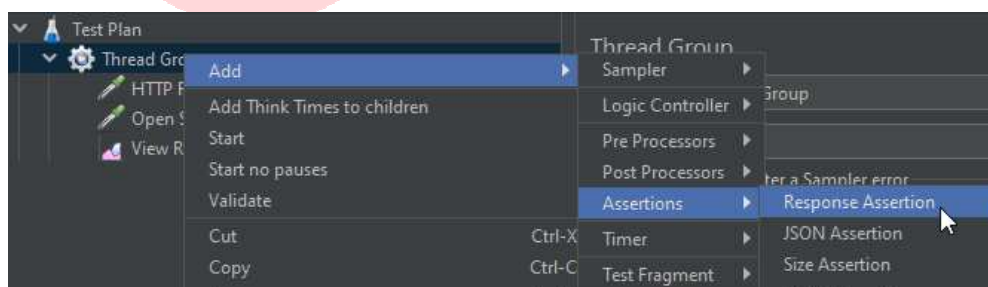
## JMeter Assertion

Kako sada radimo sa celim stranicama sajta i preuzimamo kod kao da stvarno pristupamo sajtu putem pregledača, možemo npr. proveravati da li se korisniku prikazao odgovarajući sadržaj. Pogledajmo sledeći primer. Na stranici Store, imamo link do stranice Heropedia; želimo da proverimo da li se taj link prikazao korisniku.



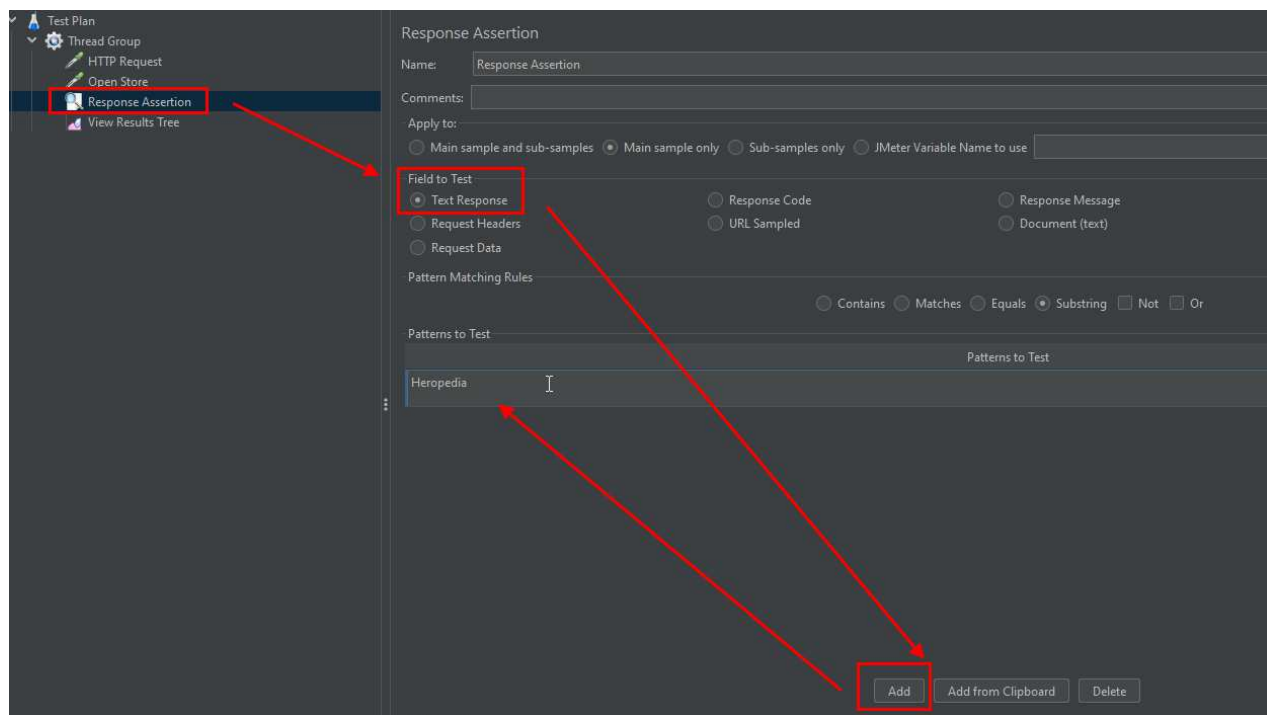
Slika 13.16. Link u okviru sajta Dota2

Ovo takođe možemo postići u okviru JMetera, korišćenjem sekcije *Assertion* i konkretno *Response Assertion*, pomoću kojih proveramo sadržaj koji dolazi u JMeter u vidu odgovora aplikacije ili web sajta.



Slika 13.17. Dodavanje Response Assertion

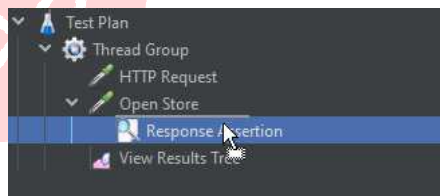
Kada dodamo *Response Assertion*, podrazumevano je postavljeno da testiramo *Text* odgovora, što i želimo, jer proveravamo reč u okviru sajta. U okviru sekcije *Patterns to Test*, biramo *Add* i u novo polje unosimo pojam čije pojavljivanje pretpostavljamo.



Slika 13.18. Dodavanje parametara *Response Assertion*a

Ostaje nam samo da povežemo pretpostavku sa odgovarajućom stranicom, jer želimo da proverimo sadržaj Store stranice. Ukoliko ne bismo izvršili povezivanje sa pravilnom stranicom, proverile bi se obe stranice. Za prvu bi se pojavila greška, a za drugu pravilan rezultat, što bi onda predstavljalo našu grešku prilikom pisanja testa, s obzirom na to da nema potrebe proveravati News stranicu za link ka stranici Heropedia.

Povezivanje postićemo jednostavnim prevlačenjem Assertiona na HTTP Request za prodavnicu:

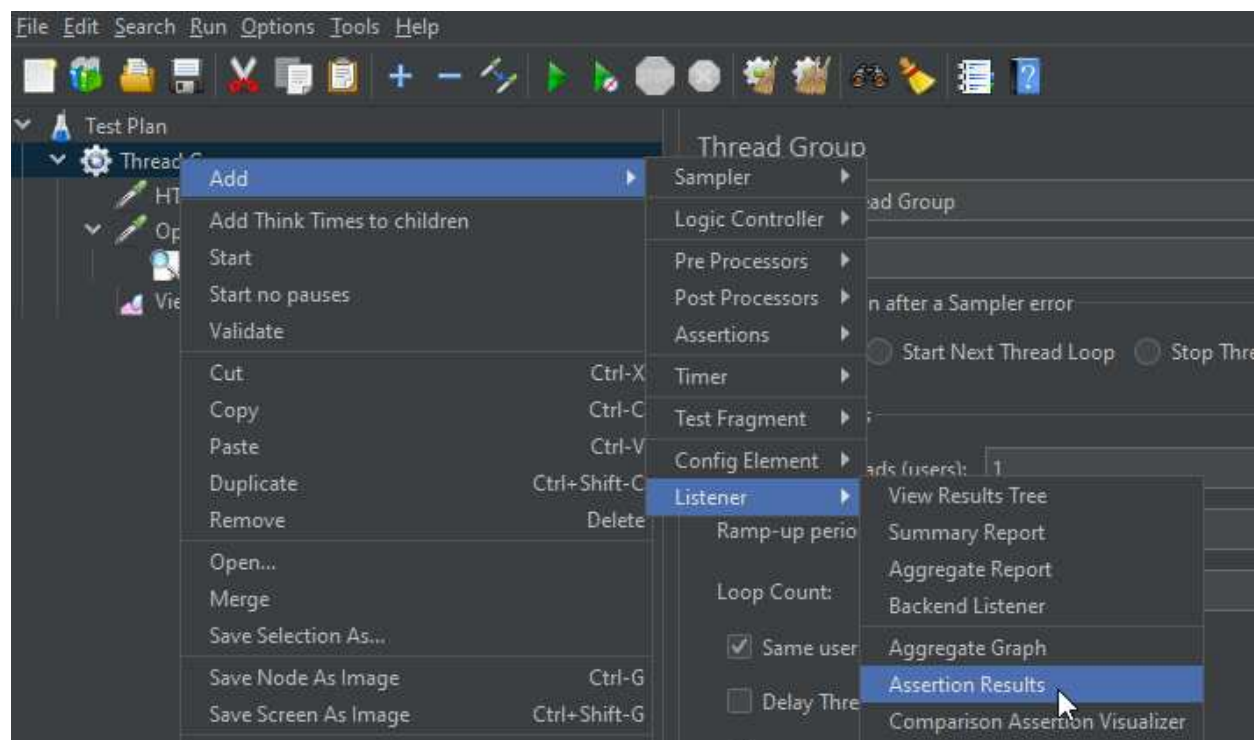


Slika 13.19. Povezivanje Assertiona sa odgovarajućim zahtevima

Kada se pojavi strelica pored naziva Requesta, znamo da smo pravilno povezali proveru i sam zahtev.

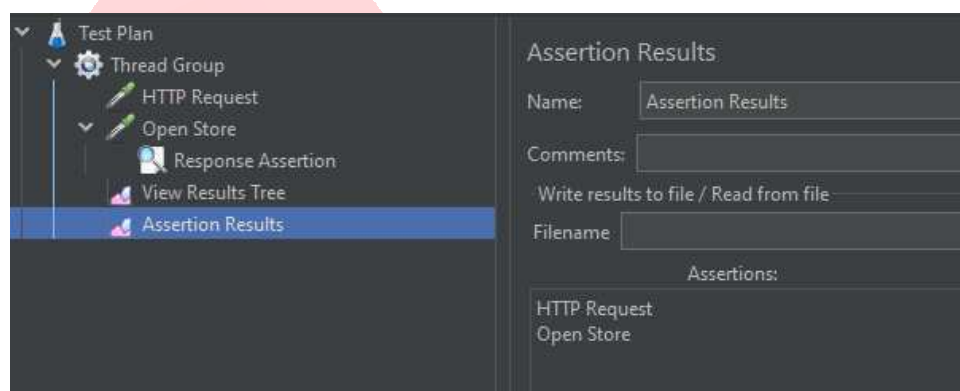
Sada je potrebno da dodamo još jedan način na koji prikazujemo rezultat. Ukoliko koristimo View Result Tree, nećemo dobiti prikaz rezultata Assertion provere sadržaja.

Stoga, dodaćemo novi način da prikazemo rezultat – to je opcija *Assertion Results*, koju možemo pronaći kao jedan od slušača.



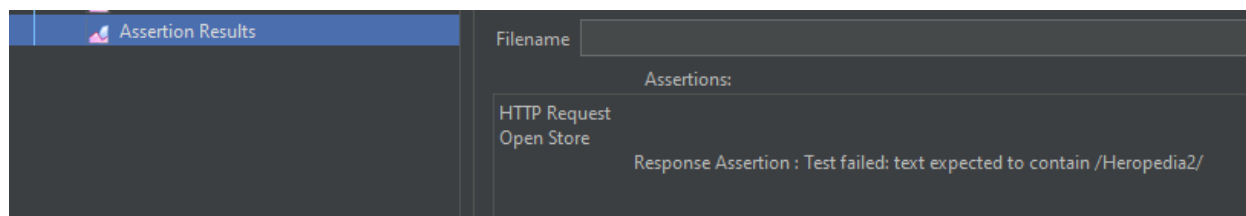
Slika 13.20. Dodavanje *Assertion Results* slušača

Ukoliko sada izaberemo novododati slušač i pokrenemo projekat, videćemo da ne dobijamo ništa u prikazu osim naziva zahteva. Ovo je ujedno dobar znak, jer se nije pojavio ispis greške na ovoj stranici.



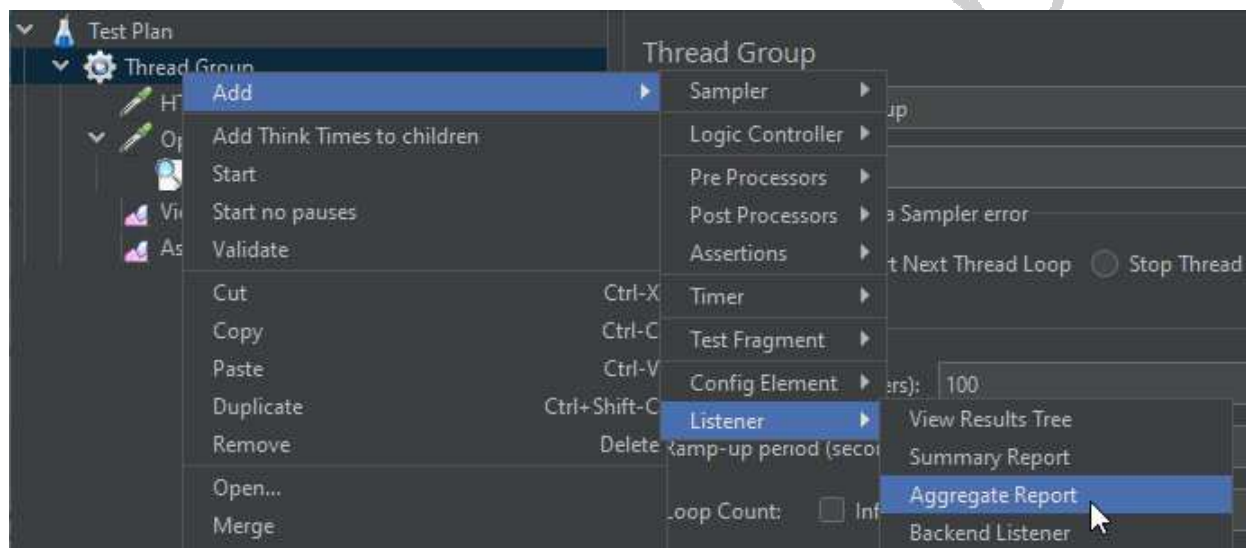
Slika 13.21. Prikaz rezultata *Assertions* sekcije

U slučaju da ne postoji sadržaj na stranici, nakon izvršenja zahteva dobili bismo informaciju o greškama koje su se javile. U našem slučaju promene parametra sa Heropedia na Heropedia2, to bi moglo da izgleda ovako:



Slika 13.22. Prikaz neuspešnog testa

Za kraj ove lekcije, prikazaćemo još jedan od načina prikazivanja rezultata na koje možemo sumirati rad naše aplikacije, a to je *Aggregate Report*.



Slika 13.23. Dodavanje Aggregate Report slušača

Nakon izbora ovog novog slušača i pokretanja projekta, dobijamo prikaz sličan sledećem:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
HTTP Request	101	587	659	764	786	824	328	872	0.00%	2.6/sec	14.68	0.33
Open Store	101	1694	1735	2141	2865	3196	580	3306	100.00%	2.5/sec	245.85	0.30
TOTAL	202	1141	804	1935	2141	3113	328	3306	50.00%	4.9/sec	255.62	0.61

Slika 13.24. Dodavanje Aggregate Report slušača

Ovaj prikaz vam sada već može biti poznat, s obzirom na to da smo slične informacije imali i u okviru Apache Bench alata, gde dobijamo statističke podatke o radu aplikacije. Na kraju, sve se svodi upravo na te podatke. Kako je JMeter prvenstveno alat za stress i load testiranje, informacije o brzinama konektovanja i preuzimanja podataka, broju grešaka i prosečnim vrednostima za svaku od operacija su upravo indikatori kvaliteta naše aplikacije pod opterećenjem.

Kao što ste imali priliku da vidite, Jmeter, u odnosu na prethodno spomenuti alat, donosi mnogo naprednije opcije i realniji pristup testiranju web aplikacija, jer nam omogućuje da simuliramo interakciju korisnika sa našim programom.

## Rezime

- Apache JMeter je open source alat napisan u Javi koji služi prvenstveno za automatsko load i stress testiranje; pored toga, ovim alatom se može pratiti funkcionalno ponašanje programa i vršiti precizno merenje performansi programa.
- Originalna svrha alata Apache JMeter je bilo testiranje web aplikacija ili FTP aplikacija. Danas se koristi za stress, load, functional i database testiranje, kao i mnoge druge vidove testiranja.
- Postoje mnogobrojne prednosti korišćenja JMeter alata: besplatan je, ima jednostavan interfejs i omogućava nam daleko bolju vizuelizaciju rezultata testiranja. Platformski je nezavisan i podržava sve moderne operativne sisteme.
- Da bi JMeter test mogao da bude pokrenut, on mora postojati na fajl sistemu, u formi test plana. Test plan se čuva u .jmx formatu. Ovo je format specifičan za Jmeter; na njega možemo gledati kao na radni fajl ili projekat u okviru JMetera. U okviru tog fajla se čuvaju svi napisani kodovi i parametri testa.
- Dodavanje korisnika u okviru JMetera se obavlja dodavanjem Thread Group, tj. grupe niti. Nakon dodavanja grupe u projekat, definišemo podešavanja o broju konkurentnih korisnika, periodu uhodavanja testa, broju iteracija testa i slično.
- Sempleri su konkretne operacije koje JMeter izvršava u okviru jedne grupe niti. Sempleri su najvećim delom mrežne operacije. Najkorišćeniji sampler je HTTP Request. HTTP Request predstavlja jedan HTTP zahtev za pristup stranici sajta.
- Listener ili slušač služi za prikaz rezultata izvršavanja semplera u okviru grupa niti. Slušači mogu vizuelizovati rezultate, transformisati ih, ili ih prosleđivati drugim sistemima (fajl sistemu, bazi podataka i slično).

